

**METHODS FOR SUPPLYING CRYPTOGRAPHIC ALGORITHM CONSTANTS
TO A STORAGE-CONSTRAINED TARGET**

TECHNICAL FIELD

The invention relates generally to security systems and, more particularly, to a security system concerning the validation of received messages.

BACKGROUND

In conventional processing systems, storage space in computer systems can be a concern. In computer systems implementing security protocols, the algorithm for security can occupy less space than the constants used to implement the security. For instance, the chain security algorithm can be implemented in less than 200 bytes, while 1024 publicly-known bytes are needed for the constants to run the chain algorithm. In the chain algorithm, the source and target systems both know and utilize the same secret key. Similarly, an algorithm such as the Secure Hash Algorithm (SHA) can be implemented in 512 bits, but needs 80 64-bit publicly known constants to make it work properly, although the source and target systems both know and utilize the same secret key in addition to the publicly known constants.

However, space constraints can create a problem with the target computer where the algorithms are to be implemented. The target computer might have memory space to store the algorithm and the secret key, but does not have all of the memory space necessary to store the publicly known constants. Increasing the size of the memory may not be a viable option, due to cost of implementation.

Therefore, there is a need for a method and a system for a target computer to process the publicly known security constants in a manner that accommodates the target

computer's memory constraints.

SUMMARY OF THE INVENTION

The present invention provides for authenticating a
5 message. A security function is performed upon the message.
The message is sent to a target. The output of the security
function is sent to the target. At least one publicly known
constant is sent to the target. The received message is
10 authenticated as a function of at least a shared key, the
received publicly known constants, the security function,
the received message, and the output of the security
function.

BRIEF DESCRIPTION OF THE DRAWINGS

15 For a more complete understanding of the present
invention, and the advantages thereof, reference is now made
to the following Detailed Description taken in conjunction
with the accompanying drawings, in which:

FIGURE 1 schematically depicts a system for
20 transferring a message, publicly known constants, and a
message authentication code (MAC); and

FIGURE 2 schematically depicts a system for
transferring a message, publicly known constants, and a CBC-
MAC value.

25

DETAILED DESCRIPTION

In the following discussion, numerous specific details
are set forth to provide a thorough understanding of the
present invention. However, those skilled in the art will
30 appreciate that the present invention may be practiced
without such specific details. In other instances, well-
known elements have been illustrated in schematic or block
diagram form in order not to obscure the present invention

in unnecessary detail. Additionally, for the most part, details concerning network communications, electro-magnetic signaling techniques, and the like, have been omitted inasmuch as such details are not considered necessary to
5 obtain a complete understanding of the present invention, and are considered to be within the understanding of persons of ordinary skill in the relevant art.

In the remainder of this description, a processing unit (PU) may be a sole processor of computations in a device.
10 In such a situation, the PU is typically referred to as an MPU (main processing unit). The processing unit may also be one of many processing units that share the computational load according to some methodology or algorithm developed for a given computational device. For the remainder of this
15 description, all references to processors shall use the term MPU whether the MPU is the sole computational element in the device or whether the MPU is sharing the computational element with other MPUs, unless otherwise indicated.

It is further noted that, unless indicated otherwise,
20 all functions described herein may be performed in either hardware or software, or some combination thereof. In a preferred embodiment, however, the functions are performed by a processor, such as a computer or an electronic data processor, in accordance with code, such as computer program
25 code, software, and/or integrated circuits that are coded to perform such functions, unless indicated otherwise.

Turning now to FIGURE 1, disclosed is a computer security system 100. In FIGURE 1, illustrated is a source computer 110 and a target computer 120 communicating across
30 an unsecured medium 115. The source computer 110 and the target computer 120 share the same secret key 125. Generally, the system 100 sends publicly known constants for a security algorithm from the source computer 110 to the

target computer 120, thereby allowing for the target computer 120 not to have these constants stored within the target computer while still allowing the target computer 120 to validate received messages. For example, SHA-512 has
5 eighty 64-bit publicly known constants which are the first 64 bits of the fractional parts of the cube roots of the first eighty prime numbers. In the system 100, the shared key 125, the security algorithm 127, the messages and the publicly known constants 129 are stored in memory of the
10 source computer 110, or are otherwise generated in a processor of the source computer 110. The shared key 125, the security algorithm 127 are stored in memory of the source computer 110, or are otherwise generated in a processor of the target computer 120.

15 In FIG. 1, source computer 110 has a message it selects to send to target computer 120 over the insecure medium 115 in a data transfer 130. The target computer 120 has the security algorithm 127 and the secret key 125, but not the publicly known constants 129 necessary for proper use of the
20 security algorithm. Therefore, the source 110 also sends the publicly known security constants 129. This occurs as follows, and is done in association with sending security-type information concerning the message itself to the target computer 120 over the one-way unsecured medium 115 in the
25 data transfer 115.

In one embodiment of the use of the security algorithm 127 in the system 100, the message from the source is hashed, a form of the security algorithm 127, using the secret key 125 and the publicly known constants 129. The
30 hashing creates a message authentication code (MAC) from the message, the secret key 125, and the publicly known constants 129. In other words, MAC is equal to a Hash function of the secret key 125, public constants 127, and

the message to be hashed. The hash algorithm can be a cryptographically secure one-way hash function, such as the SHA, for the Secure Hash Algorithm (SHA).

After performing the security function, the source
5 computer 110 then sends the Message (perhaps encrypted), the MAC value, or some other security value and the publicly known constants 129 used in the hash to the target computer 120 over the unsecured medium 115 in the data transfer 130. The target computer 120 receives the computed MAC or other
10 security code, the message and the publicly known constants 129.

The target computer 120 then recomputes its own MAC value, using its secret key 125, the received publicly-known constants 129 from the source computer 110, the received
15 message, the shared key 125, and the hash function or other security algorithm 127. If the MAC the target computer 120 calculates for the received message using the received publicly known constants 129 and the security algorithm 127 and the secret key 125 equals the MAC the target computer
20 120 received from the source 110, then the message and the constants are authentic.

If the MAC calculated by the target computer 120 and the MAC received by the source computer 110 do not equal, either the message or the publicly known constants 129 have
25 been changed, and therefore the Message is not validated as authentic.

Note that an unauthorized person could read or modify the Message, the MAC or the hash constants when these values are conveyed over the one-way unsecured medium 115 and try
30 to "fool" the target computer 120 as to the authenticity of the communication. However, the MAC is a function of the message, the publicly known constants 129 constants, and the secret key 125, and the security algorithm 127. As the

unauthorized will not know what the secret key 127 is, it is highly unlikely that an adversary would be able to properly change the message and the publicly known constants 129 and the message to pass authentication by the target computer
5 120.

Turning now to FIGURE 2, disclosed is a computer security system 200 that employs encryption for the sent message for added security for the message. In FIGURE 2, a source computer 210 and a target computer 220 communicating
10 across an unsecured medium 215. The source computer 210 and the target computer 220 share the same secret key 225. Generally, the system 200 sends publicly known constants for a security algorithm from the source computer 210 to the target computer 220, thereby allowing for the target
15 computer 220 not to have these constants stored within the target computer while still allowing the target computer 220 to validate received messages.

In the system 200, the shared key 225, the security algorithm 227, the messages and the publicly known constants
20 129 are stored in memory of the source computer 210, or are otherwise generated in a processor of the source computer 210. The shared key 225, and the security algorithm 227 are stored in memory of the target computer 210, or are otherwise generated in a processor of the target computer
25 220.

The target computer 220 performs an optional decryption upon the message from the source computer 110, as well as a validation of the received message. In one embodiment, the source computer 210 uses CBM-MAC (cipher
30 block chaining message authentication code) as its security algorithm 227 on the message before sending the encrypted message and the publicly known constants 229 to the target computer 220.

When performing the encryption, the source computer 220 can perform the encryption substantially as follows, although those of ordinary skill in the art, other encryption algorithms are within the scope of the present
5 Application.

First, the message to be sent is broken down into a series of blocks $M = M_1, M_2 \dots M_n$. Then, values useful for running the security algorithm 127 are generated. First, $Y_1 = \text{Enc}(M_1)$. In other words, the value Y_1 is created from the
10 encryption of M_1 , using the secret key value 225, the security algorithm 227 and the publicly known constants 229.

Furthermore, there are other Y values created, $Y_2, Y_3, Y_4 \dots Y_n$. These other values are created as follows. $Y_i = \text{Enc}(M_i \text{ XOR } Y_{i-1})$. In other words, the Y_i value is the
15 encryption of the M_i value "Exclusive-Or"ed with the Y_{i-1} value. The Y_i value is then used when determining the Y_{i+1} value, and so on, until the last M_i value is encrypted in Y_n . Y_n is then the CBC-MAC value.

The source computer 210 then sends the message, the
20 CBC-MAC value and the publicly known constants 229 to the target computer 220. The target computer then performs its own CBC-MAC check using the received publicly known constants 229, the secret key 225, and the received message. The message can be encrypted or unencrypted. Both options
25 are allowed. If the computed CBC-MAC is the same as the received CBC-MAC, then the target 220 determines that the message and publicly known constants 220 are authentic, in other words, not modified. The target 220 uses the shared key 220, the security algorithm 227, and the received
30 message, publicly known constants 229 and the CBC-MAC value for computing its own CBC-MAC.

Although the CBC-MAC, the message itself, and/or the constants could be altered during transmission over the

insecure media, it is highly unlikely that an adversary would be able to successfully make the correct changes to the CBC-MAC, the constants and/or the message, because the adversary would not know the secret key to do this properly.

5 Note that while the message itself can be read in the unsecured environment, it could not be altered. For some kinds of security applications, this is sufficient.

It is understood that the present invention can take many forms and embodiments. Accordingly, several variations
10 may be made in the foregoing without departing from the spirit or the scope of the invention. The capabilities outlined herein allow for the possibility of a variety of programming models. This disclosure should not be read as preferring any particular programming model, but is instead
15 directed to the underlying mechanisms on which these programming models can be built.

Having thus described the present invention by reference to certain of its preferred embodiments, it is noted that the embodiments disclosed are illustrative rather
20 than limiting in nature and that a wide range of variations, modifications, changes, and substitutions are contemplated in the foregoing disclosure and, in some instances, some features of the present invention may be employed without a corresponding use of the other features. Many such
25 variations and modifications may be considered desirable by those skilled in the art based upon a review of the foregoing description of preferred embodiments. Accordingly, it is appropriate that the appended claims be construed broadly and in a manner consistent with the scope
30 of the invention.